UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/734,959	12/12/2003	Tin-Fook Ngai	20002/17846	4798

34431 7590 03/05/2008

HANLEY, FLIGHT & ZIMMERMAN, LLC
150 S. WACKER DRIVE
SUITE 2100
CHICAGO, IL 60606

EXAMINER	
CHOU, ANDREW Y	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
03/05/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/734,959	NGAI ET AL.	
	Examiner	Art Unit	
	ANDREW CHOU	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11/26/2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the remarks filed on 11/26/2007.

2. Claim 33 is new. Claims 1-33 are still pending.

Response to Arguments

3. Applicant's arguments with respect to claims rejection have been considered but are moot in view of the new ground(s) of rejection See "The Structure of a Compiler for Explicit and Implicit Parallelism", Seon Wook Kim and Rudolf Eigenmann, 2001 (hereinafter Kim) art made of record as applied below.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-33 are rejected under 35 U.S.C. 102(b) as being anticipated by "The Structure of a Compiler for Explicit and Implicit Parallelism", Seon Wook Kim and Rudolf Eigenmann, 2001 (hereinafter Kim).

Claim 1:

Kim discloses a method of compiling a program comprising:

identifying a set of speculative parallel thread candidates (see for example page 2, "Recognition of parallelism");

determining cost values for at least some of the speculative parallel thread candidates (see for example pages 4-5, "2.2, Thread Selection");

selecting a set of speculative parallel threads from the set of speculative parallel thread candidates based on the cost values (see for example pages 4-5, "2.2 Thread Selection"); and generating program code based on the set of speculative parallel threads (see for example page 2, "Thread code generation").

Claim 2:

Kim further discloses a method as defined in claim 1 wherein identifying the set of speculative parallel thread candidates comprises identifying program regions (see for example page 2, "Selecting explicit and implicit threads").

Claim 3:

Kim further discloses a method as defined in claim 1 wherein at least one of the speculative parallel thread candidates comprises at least one program region (see for example page 2, "Selecting explicit and implicit threads").

Claim 4:

Kim further discloses a method as defined in claim 1 wherein at least one of the speculative parallel threads comprises at least one program region (see for example page 2, "Selecting explicit and implicit threads").

Claim 5:

Kim further discloses a method as defined in claim 1 wherein identifying the set of speculative parallel thread candidates comprises identifying program loops (see for example pages 4-5, "2.2 Thread Selection").

Claim 6:

Kim further discloses a method as defined in claim 1 wherein at least one of the speculative parallel thread candidates comprises a program loop (see for example pages 4-5, "2.2 Thread Selection").

Claim 7:

Kim further discloses a method as defined in claim 1 wherein at least one of the speculative parallel threads comprises a program loop (see for example pages 5-6, "2.3 Thread Preprocessing").

Claim 8:

Kim further discloses a.method as defined in claim 1 wherein identifying the set of speculative parallel thread candidates comprises identifying a main thread (see for example pages 5-6, "2.3 Thread Preprocessing").

Claim 9:

Kim further discloses a method as defined in claim 8 wherein the main thread comprises a current iteration of a program loop, and the speculative parallel thread candidate comprises a next iteration of the same program loop (see for example pages 5-6, "2.3 Thread Preprocessing").

Claim 10:

Kim further discloses a method as defined in claim 8 wherein the main thread comprises a current iteration of a program loop, and the speculative parallel thread comprises a next iteration of the same program (see for example pages 5-6, "2.3 Thread Preprocessing").

Claim 11:

Kim further discloses a method as defined in claim 1 wherein the cost value is a misspeculation cost (see for example pages 4-5, "2.2 Thread Selection").

Claim 12:

Kim further discloses a method as defined in claim 11 wherein determining the misspeculation cost comprises: identifying a data dependency in the speculative parallel thread candidate (see for example page 6, paragraph 2, "The first type of dependence...."); determining, for the data dependency, a likelihood that a dependency violation will occur; and determining an amount of computation required to recover from the data dependency violation (see for example page 6, paragraph 2).

Claim 13:

Kim further discloses a method as defined in claim 1 further comprising determining at least one of the following for at least one of the speculative parallel thread candidates: a Size of the speculative parallel thread candidate; and a likelihood representative of the speculative parallel thread candidate (see for example page 6, Fig. 2, and related text).

Claim 14:

Kim further discloses a method as defined in claim 1 wherein at least one of the

speculative parallel thread candidates is transformed prior to determining the cost value for the at least one of the speculative parallel thread candidates (see for example page 6, Fig. 2, "Code transformation...", and related text).

Claim 15:

Kim further discloses a method as defined in claim 14 wherein the at least one of the speculative parallel thread candidates is transformed by a code reordering (see for example page 6, Fig. 2, "Code transformation...", and related text).

Claim 16:

Kim further discloses a method as defined in claim 14 further comprising determining at least one of the following for at least one of the speculative parallel thread candidates: a size of the speculative parallel thread candidate; a likelihood representative of the speculative parallel thread candidate (see for example page 6, Fig. 2, and related text); and a description of the transformation performed on the speculative parallel thread candidate (see for example page 6, Fig. 2, and related text).

Claim 17:

Kim further discloses a method as defined in claim 1 wherein at least one of the speculative parallel threads is transformed prior to code generation (see for example page 6, Fig. 2, "Code transformation...", and related text).

Claim 18:

Kim further discloses a method as described in claim 17 wherein the at least one of the speculative parallel threads is transformed by code reordering (see for example page 6, Fig. 2, "Code transformation...", and related text).

Claim 19:

This is the article version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Kim.

Claim 20:

This is the article version of the claimed method discussed above (Claim 11), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Kim.

Claim 21:

This is the article version of the claimed method discussed above (Claim 12), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Kim.

Claim 22:

This is the article version of the claimed method discussed above (Claim 13), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also.anticipated by Kim.

Claim 23:

This is the article version of the claimed method discussed above (Claim 14), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Kim

Claim 24:

Kim discloses an apparatus (see for example Fig. 1, and related text) to compile a program comprising: a candidate identifier to identify a set of speculative parallel thread candidates (see for example page 2, "Recognition of parallelism"); a metric estimator to determine a cost value for at least one of the speculative parallel thread candidates (see for example page 6, Fig. 2, and related text); a speculative parallel thread selector, to select a set of speculative parallel threads from the set of speculative parallel thread candidates based on the cost values (see for example page 2, "Selecting explicit and implicit threads", and related text); and a code generator to generate program code based on the set of speculative parallel threads (see for example page 4, Fig. 1, "Code generator", and related text).

Claim 25:

Kim further discloses a apparatus as defined in claim 24 wherein the candidate identifier comprises a region identifier to identify program regions (see for example page 2, "Recognition of parallelism").

Claim 26:

Kim further discloses a apparatus as defined in claim 24 wherein the candidate identifier comprises a loop identifier to identify program loops (see for example pages 4-5, "2.2 Thread Selection").

Claim 27:

Kim further discloses a apparatus as defined in claim 24 wherein the candidate identifier comprises a candidate selector to select a first one of a program region and a program loop iteration to execute in a main thread, and to select a second one of a program

region and a program loop iteration to execute in a speculative parallel thread (see for example pages 4-5, "2.2 Thread Selection").

Claim 28:

Kim further discloses a apparatus as defined in claim 24 wherein the metric estimator determines a misspeculation cost (see for example pages 4-5, "2.2 Thread Selection").

Claim 29:

Kim further discloses a apparatus as defined in claim 24 wherein the metric estimator comprises: a data dependency identifier to identify a data dependency in the speculative parallel thread candidate (see for example page 6, paragraph 2, "The first type of dependence...."); a likelihood evaluator to determine a likelihood that a dependency violation will occur (see for example page 6, paragraph 2, "The first type of dependence...."); and a recovery size calculator to determine an amount of computation required to recover from the data dependency violation (see for example page 6, Fig. 2, and related text).

Claim 30:

Kim further discloses an apparatus as defined in claim 24 wherein the candidate identifier determines at least one of the following for at least one of the speculative parallel thread candidates: a size of the speculative parallel thread candidate violation (see for example page 6, Fig. 2, and related text); and a likelihood representative of the speculative parallel thread candidate violation (see for example page 6, Fig. 2, and related text).

Claim 31:

Kim discloses a system (see for example page 4, Fig. 1, and related text) to compile a program comprising: a candidate identifier to identify a set of speculative parallel thread candidates (see for example page 2, "Recognition of parallelism"); a metric estimator to determine a cost value for at least one of the speculative parallel thread candidates (see for example page 6, Fig. 2, and related text); a speculative parallel thread selector, to select a set of speculative parallel threads from the set of speculative parallel thread candidates based on the cost values (see for example page 2, "Selecting explicit and implicit threads", and related text); and a code generator to generate program code based on the set of speculative parallel threads (see for example page 4, Fig. 1, "Code generator", and related text).

Claim 32:

Kim further discloses a system as define in claim 31 wherein the metric estimator comprises:

a data dependency identifier to identify a data dependency in the speculative parallel thread candidate (see for example page 6, paragraph 2, "The first type of dependence...."); a likelihood evaluator to determine a likelihood that a dependency violation will occur (see for example page 6, paragraph 2, "The first type of dependence...."); and a recovery size calculator to determine an amount of computation required to recover from the data dependency violation (see for example page 6, Fig. 2, and related text).

Claim 33:

Kim discloses a method of compiling a program comprising:

identifying a data dependency violation in a set of speculative parallel thread candidates (see for example pages 6-7, "The second type of dependence...."); determining a likelihood that the data dependency violation will occur (see for example pages 9-10, "Implicit/Explicit Thread Selection"); determining an amount of computation required to recover from the data dependency (see for example page 5, "Two-version Code for Explicit/Implicit Threading"); and selecting at least one of the set of speculative parallel thread candidates based on a lowest likelihood of misspeculation (see for example pages 9-10, "Implicit/Explicit Thread Selection").

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andrew Y. Chou whose telephone number is (571) 272-6829. The examiner can normally be reached on Monday-Friday, 8:00 am - 4:30 pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached on (571) 272-3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published

applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair- direct.uspto.,qov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll free)

AYC

TUAN DAM
SUPERVISORY PATENT EXAMINER